

1. 2016-2

```
001 /* (c) 2016 Rahmat M. Samik-Ibrahim -- This is free software
005  * Assume (&ptrchr is 0x7FFFEEDDCCBB, order of bytes: little-endian) */
009 #define LINES  3
010 #include <stdio.h>
012 void printeq(int lines) {
013     while (lines-- > 0 ) printf(" = ");
014     printf("\n");
015 }
017 void main(void) {
018     int          ii;
019     unsigned char dummy  = 'a';
020     unsigned char* ptrchr = &dummy;
022     printeq(LINES);
023     printf(" dummy:  %c\n", dummy);
024     printf(" *ptrchr: %c\n", *ptrchr);
025     printeq(LINES);
026     printf("%p\n", &ptrchr);
027     printeq(LINES);
028     ptrchr = (char*) &ptrchr;
029     for (ii=0; ii<6; ii++) {
030         printf("%X ", *ptrchr);
031         ptrchr++;
032     }
033     putchar('\n');
034     printeq(LINES);
035 }
```

(a) Write down the output of this program

2. 2017-1

C Programing	
001 /* 002 * (c) 2017 Rahmat M. Samik-Ibrahim 003 * -- This is free software 004 * REV00 Thu Mar 30 18:27:30 WIB 2017 005 * START Thu Mar 30 18:27:30 WIB 2017 006 * INT is 32 bit little endian 007 * 41H='A'; 42H='B'; 43H='C'; 44H='D' 008 */ 009 #include <stdio.h> 010 char chrary[]="ZZZZ ZZZZ ";	011 void main(void) { 012 char chrvar = 'M'; 013 int intvar = 0x41424344; 014 int* intptr = (int*) chrary; 015 printf("YY. chrary=%p\n", chrary); 016 printf("ZZ. intptr=%p\n", intptr); 017 printf("01. chrvar=%c\n", chrvar); 018 printf("02. *chrvar=%c\n", *chrvar); 019 printf("03. str chrary=%s\n", chrary); 020 *intptr = intvar; 021 printf("04. str chrary=%s\n", chrary); 022 }
Program Output (Line: 015, 016, 017, 018, 019, 021):	
YY. chrary=0x600a08	

4. 2018-1

```
018 /* Clue#1: All strings end with 0x00 */
019 /* Clue#2: Address=64 bit BIG ENDIAN */
020 /* Clue#3: ASCII '0' (Zero) is 0x30 */
021 /* Clue#4: ASCII 'A' is 0x41 */
```

```
001 /* (c) 2018 This is a free program */
002 /* Rahmat M. Samik-Ibrahim */
003 /* The "array" starts at 0x601040 */
004 /* The "pointer" address is 0x601050 */
005
006 #include <stdio.h>
007
008 char array[]="0123456789ABCDE";
009 char* pointer=array;
010 void main(void) {
011     printf("START\n");
012     printf("%p\n", &pointer);
013     printf("%p\n", pointer);
014     printf("%s\n", pointer);
015     printf("%d\n", pointer[15]);
016     printf("STOP\n");
017 }
```

Initially, addresses 0x601040 - 0x60105F = 0x00. What will be in those addresses after executing the program?

[illegible]

[illegible]

[illegible]